Steroid and Non-Steroid Bodybuilder Classification Using Deep-Learning

Safa Ahmed
Department of Electrical and Computer Engineering, North South University
Dhaka,Bangladesh
safa.ahmed@northsouth.edu

Md. Tahmid Obayed
Department of Electrical and Computer Engineering, North South University
Dhaka,Bangladesh
tahmid.obayed@northsouth.edu

Dr. Mohammad Monirujjaman Khan
Department of Electrical and Computer Engineering, North South University
Dhaka,Bangladesh
monirujjaman.khan@northsouth.edu

Abstract—**This paper aims to implement an image classification system to detect Steroid and Non-steroid bodybuilders with RGB images using deep learning techniques. The purpose of creating a steroid use detection Deep Learning system is to provide a reliable dope-testing alternative for sports organization committees and athletes without the need for expensive time-consuming lab tests that can easily be bribed out of. Moreover, this allows a method to avert the failure of detecting steroid use in bodybuilders, as its use leads to unfair competitive advantages and life-threatening health risks. Particularly, cardiovascular diseases like high blood pressure and stroke, infections like hepatitis, and HIV, and psychiatric effects like mania, and addiction. While some side effects may be semi-permanent and reversible once the drug use is stopped some leave irreparable life-long damage. Therefore, we have created a CNN-based Deep-learning steroid detection system that doesn't rely on people but on data and proof. Firstly, a non-steroid and steroid bodybuilder image dataset of 4,231 RGB images was collected and cleaned. Next, we trained multiple pre-trained CNN models using this dataset. Lastly, we have reached reliable results with our pre-trained models, namely, our highest accuracy being 98.00% with InceptionV3 of epoch 30, and our lowest accuracy is 94.00% with Resnet50 of epoch 50, all other model results fall into this range. Finally, this is how our project provides an accurate Deep-learning Image classification system capable of steroid and non-steroid bodybuilder detection.**

Keywords—Deep Learning, Convolutional Neural Network (CNN), Image Classification, Pre-Trained Models, Machine Learning

I. Introduction

The categorization of steroid and non-steroid bodybuilders is a significant obstacle for sports medicine and anti-doping activities. Steroid use occurs discreetly and is stigmatized or even forbidden in certain nations. Therefore, providing accurate information on steroid usage in bodybuilding is both challenging and important.

However, according to 2023 research by the International Society of Sports Nutrition (ISSN), 3.3% of male and 0.5% of female bodybuilders used anabolic-androgenic steroids (AAS). AAS was also reported to be used by around 6% of American male high school students [1, 2].

Furthermore, steroid usage can impact our health by causing liver damage, cardiovascular disease, and mental health issues. Users usually develop addictions, body dysmorphia issues, and overall portray a false narrative of what a muscular body looks like to the public and especially young sports enthusiasts. Additionally, utilizing steroids gives sports players an unfair competitive advantage that leads to a pattern of unfair winning.

To combat such a global health issue in the world of sports and sports medicine Deep Learning techniques could be explored. Since deep learning approaches have been successfully used for several classification challenges in multiple projects all over the world this would be a reliable technique. Furthermore, various research studies have already been in the works, including the classification of Steroid and Non-steroid bodybuilder images [3, 4]. Deep learning has the capacity to understand complex data structures, which will provide predictions using pre-trained models, resulting in the achievement of the objective.

Our goal with this project is to introduce an easier testing process for differentiating Steroid and Non-steroid bodies by the use of transfer learning techniques. By taking the images of the athletes' bodies and testing them against our model we can show whether the athletes use steroids or not.

Since there is also no proper steroid and non-steroid bodybuilder dataset for it yet, we have created a dataset by collecting and cleaning 4,231 RGB images, trained various pre-trained CNN models with it and created a primary User Interface using StreamLit for customer use.

II. Related Research

A. Study on Steroids

According to a study conducted by Zeeuw and colleagues (2023), the use of anabolic androgenic steroids (AAS) is a significant public health concern due to its negative consequences and causes of AAS use disorder (AASUD) [1]. The study found that the frequency, duration, and average dosage of AAS strongly affect the rise of AASUD. Additionally, the simultaneous use of other psychoactive substances can worsen the negative consequences of AAS use, including an increased risk of AAS dependence. Furthermore, the study discovered that AAS use is linked to a range of mental health disorders, such as depression, anxiety, and bipolar disorder. Individuals with AASUD have a higher chance of symptoms like depression and anxiety. The authors also noted that AAS users with AASUD are more likely to have a history of psychiatric hospitalization and a family history of mental health disorders. These findings highlight the importance of targeted interventions to reduce AAS use and AASUD, while also highlighting simultaneous substance use and fulfilling the mental health needs of AAS users. Mazzeo et al. has stated that nabolic steroids, usually taken orally or through injections, are used to improve physical outlook and athletic performance in athletes [2]. But long-term abusers may face irreversible psychological and behavioral changes. Moreover, this increases the risk of multiple serious health concerns such as sterility, cardiovascular diseases, and even liver cancer. Even though the International Olympic Committee banned its use, athletes and bodybuilders still widely use anabolic steroids hence, reinforcing the importance of education on its risks.

B. Concerning Convolutional Neural Networks (CNN)

Liew et al. proposed an optimized CNN architecture for gender classification. The architecture consists of fused convolutional and subsampling layers, and cross-correlation is applied in the processing layers instead of convolution [3]. Benchmarking results have shown that the proposed CNN has superior classification performance, achieving 98.75% and 99.38% classification rates on the SUMS and AT&T face databases respectively. Harakannanavar et al. conducted a study using machine learning and image-processing techniques to identify leaf illnesses in tomato plants [4]. The authors used samples of tomato leaves with diseases to develop their model and suggested that farmers could use these samples to detect infections based on early signs. They down sampled the tomato leaf samples to 256x256 pixels and applied histogram equalization to enhance their quality. The authors introduced K-means clustering to divide the data space into Voronoi cells and used contour tracing to retrieve the boundary of the leaf sample. They extracted relevant characteristics of the leaf samples using descriptors such as Discrete Wavelet Transform, Principal Component Analysis, and Grey Level Co-occurrence Matrix. Harakannanavar et al. employed machine learning techniques such as Support Vector Machine, Convolutional Neural Network, and K-Nearest Neighbour to classify the retrieved features. They used tomato-disordered samples to assess the accuracy of the proposed model, with SVM achieving 88%, K-NN achieving 97%, and CNN achieving 99.6% accuracy.

C. Pursuing Pre-trained Models

This study by K. He et al. cites Densely Connected Convolutional Networks where authors proposed that, instead of the traditional one-layer-to-one-layer connections, DenseNet takes a bold step, connecting each layer to every other layer [20]. Imagine the camaraderie L(L+1)/2 direct connections for L layers! By sharing information generously between layers, DenseNet tackles the vanishing-gradient issue, ramps up feature propagation, encourages feature recycling, and does all of this while significantly cutting down on the number of parameters. DenseNet not only outshines its peers but does so with less computational heavy lifting. This paper cites DenseNet's architecture but also the powerful performance boost it brings to the neural network. A. Qamar Bhatti et al, presented an innovative solution to the problems associated with deep neural network depth escalation in their ground-breaking research on Deep Residual Learning for Image Recognition [12]. This article specifically references layer inputs to reformulate layers into residual functions, citing ResNet-50's unique features. This clever approach not only simplifies the optimization process but also allows networks to achieve remarkable depth—152 layers on the ImageNet dataset—beating less complicated but shallow designs such as VGG. The results demonstrate the exceptional effectiveness of residual networks, with an error rate of only 3.57% on the ImageNet test set for classification tasks. This work explores the critical significance of representation depth in visual recognition tasks, going beyond image detection and demonstrating a significant 28% relative improvement on the COCO object detection dataset. The fundamental part ResNet-50 played in reaching this cutting-edge image recognition performance is regarded as evidence of its enduring influence. According to C. Peng et al., the research's purpose is to reduce the number of parameters and boost efficiency by scaling up networks while weighing the benefits of increased processing power, regularization, and factorizing convolutions [14]. This solution, with a computational cost of 5 billion multiply-adds and fewer than 25 million parameters, shows the strength of the model with a single frame assessment providing a 21.2% top-1 and 5.6% top-5 error. With four models combined with multi-crop assessment, this validation set has a top-1 error of 17.3% and a top-5 error of 3.5%. Therefore, 3.6% of top-5 errors for the official set indicate how state-of-the-art InceptionV3 is. M. Tan and Q. Le explore Convolutional Neural Network (ConvNet) development through model scaling and emphasizing the balance of network depth, width, and resolution for enhanced performance. By introducing a novel scaling method using a compound coefficient, the study showcases its effectiveness in scaling up MobileNets and ResNet [16]. Further innovation involves neural architecture search, resulting in EfficientNet which attains a high 84.4% top-1 / 97.1% top-5 accuracy on ImageNet, 8.4 times smaller, and 6.1 times faster than the best existing ConvNet. This efficiency extends to transfer learning, achieving top-tier accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and three other datasets, with an order of magnitude fewer parameters. This study is significant in optimizing ConvNets for superior performance within resource constraints. R. Kumar et al. found that Dog breed prediction of deep learning developed using a convolutional neural network to predict the breed of a hundred images by taking their images as input [24]. Usage transfer learning on the way to build models that make output and around to hundreds of dissimilar dog types. The algorithm identified dog breeds quite exactly and had good results. Transfer learning takes an excessive choice in the upcoming in joining a prebuilt model by the model they created. Benchmarking results have shown that the proposed CNN has superior classification performance, achieving 98.75% and 99.38% classification rates on the VGG16 and AT&T dog databases respectively.

III. Method

For the project our dataset was created consisting of Steroid and Non-steroid bodybuilder images sent through Image Pre-processing [5, 6]. After that, they are split and put into separate Training set and Validation set folders. Next, we train various pretrained models with the dataset and observe the result. Finally, in the testing phase we deploy the best model into the WebApp and test for the best predictions. The overall training process has been described below.

A. Architecture

For our project we use feature extractor, f, for InceptionV3, Resnet50, EfficientNetB0, and Densenet121. Here, the feature extractors have been pretrained on ImageNet which consists of 1000 classes. We remove the linear layer of all the feature extractors for our respective task, binary classification. We use pretrained weights for better convergence. In Figure 1 below, the detailed InceptionV3 Architecture is portrayed.

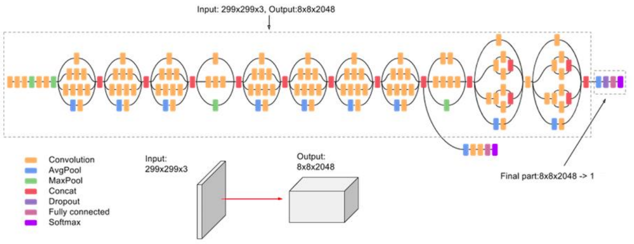

Fig. 1. InceptionV3 Architecture for our project

B. Dataset and Dataset Pre-processing

No image dataset for the categorization of Steroids and Non-steroids was available. Therefore, to use our deep learning model, we had to construct a dataset. 4,231 RGB photos total—2,076 non-steroid images and 2,155 steroid images—are included in the collection. The folder name works as a descriptor for both steroid and non-steroid photos. Next, to preserve the dataset's consistency, pre-processing first changes all our RGB photos to (224x224x3) pixels [7, 8]. Then, since CNN needs a tensor-type data structure to calculate, we do this to input our data into the pre-trained CNN model [9, 10]. After, for conducting our neural networks more efficiently, normalization puts pixel values in a standard format, which is why it's crucial for pre-processing image datasets. Here, we scale our pixel values in the standard form by taking the mean value and dividing it by the standard deviation. Lastly, to train models we split our dataset into 80% for the Training set, 10% for the Validation set, and 10% for the Test set. We performed rotation-based image augmentation based on a predefined offset on a certain amount of the training set. Next, our step is to train with various pre-trained models and observe the result. Finally, in the testing phase we deploy the best model into the WebApp and test for the best predictions. In Figure 2 below, the design of our project is illustrated.

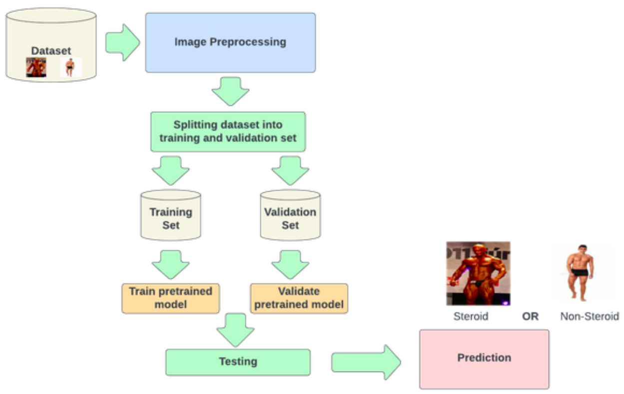

Fig. 2. Block Diagram for Steroid and Non-steroid Image Classification.

C. WebApp

After training all the pre-trained models we will implement the best model in the WebApp that yields the highest accuracy. To do that we have to save the best model in .h5 file format. Next, to create the app, we build a Python App by first importing the required libraries and storing the labels from the labels.txt file. Then, we run the app through either Localhost or Networkhost of our preferred device. On the app, we browse through the local drive and choose a file. The app then uploads the chosen file, loads the best model, and sends the file for classification. Here, the image is resized to (224, 224, 3) pixels, converted into a numpy array, normalized, set as the model's input and the predicted class name is printed as the output beside "RESULT:" on the Web page. In Figure 3 below, a screenshot of the WebApp is portrayed.

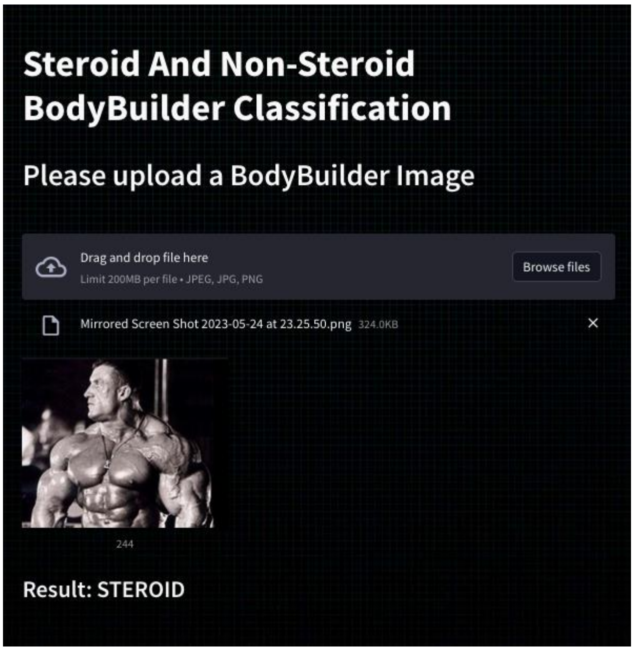

Fig. 3. Screenshot of StreamLit WebApp

IV. Result Analysis and Discussion

After training four different pre-trained models namely DenseNet121 [7, 11], ResNet50 [12, 13], InceptionV3 [14, 15], EfficientNetB0 [16, 17], we observed that the best result is achieved by InceptionV3 with an accuracy of 98.00% and lowest validation loss.

A. Resnet-50

In the study, we see that the pre-trained model Resnet50 classifies Steroids and Non-steroids images with an accuracy of 94.00%. Here, Precision of the Steroid class is 0.98 and Non-steroid class is 0.91, similarly, Recall of the Steroid class is 0.89 and Non-steroid class is 0.98 and for F1-score we see that for the Steroid class, it is 0.93 and for Non-steroid it is 0.94. To achieve these results, we must optimize our hyperparameters, by setting them to the following, Batch of 32, 50 Epochs, Learning rate of 0.001, Adam optimizer, Activation Function is Softmax, and Loss Function of Cross-Entropy Loss.

In Figure 4 below, we can see the Accuracy is 94% for Resnet-50. While, Precision of Steroid class is 0.98 and Non-steroid class is 0.91, Recall of Steroid class is 0.89 and Non-steroid class is 0.98, and F1-score of Steroid class is 0.93 and Non-steroid class is 0.94.

Classification Report:	precision	recall	f1-score	support
Steroid	0.98	0.89	0.93	206
Nonsteroid	0.91	0.98	0.94	215
accuracy			0.94	421
macro avg	0.94	0.94	0.94	421
weighted avg	0.94	0.94	0.94	421

Fig. 4. Classification report of Resnet-50

In Figure 5 below, we observe the training and validation accuracy curve of Resnet-50 which visualizes the change in Accuracy of Training and Validation on the y-axis over Epoch in the x-axis. From 10 to 30 epoch validation accuracy was fairly consistent, but from 30 epoch to 40 epoch validation acccuracy highly dropped and continued to drop. However, from 50 epoch onward validation accuracy somewhat stabalized.

Fig. 5. Training and Validation Accuracy of Resnet-50

In Figure 6 below, we observe the training and validation loss curve of Resnet-50 which visualizes the change in Loss of Training and Validation on the y-axis over Epoch in the x-axis. From 0 to 10 epoch validation loss kept rising immensely, but from 10 epoch until 30 epoch validation loss stayed faily consistent. Then, from 30 epoch onward validation loss rised against until 40 epoch. Finally from 40 to 50 epoch it started to stabalize a little.

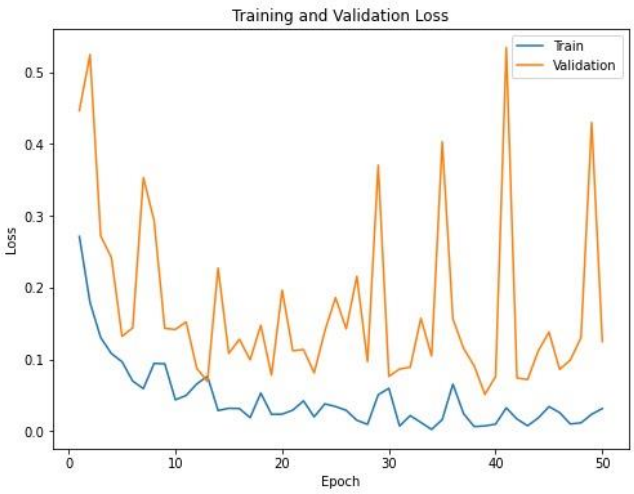

Fig. 6. Training and Validation Loss of Resnet-50

In Figure 7 below, the confusion matrix visualizes all the true and false predictions of steroids and non-steroid classes by Resnet-50. Here, 184 steroid images are predicted as steroid images and 211 non steeroid images are predicted as nonsteroid images. While, 4 nonsteroid images are predicted wrong and 22 steroid images are predicted wrong too.

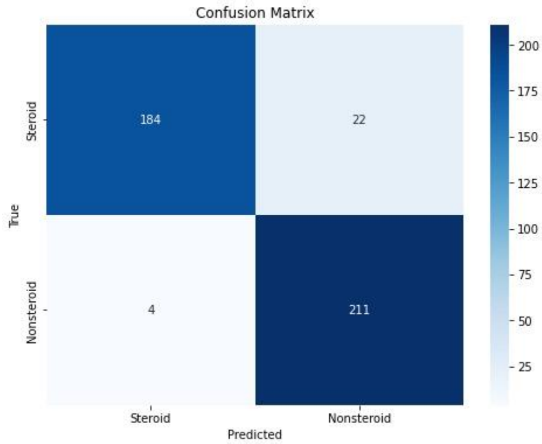

Fig. 7. Confusion Matrix of Resnet-50.

B. InceptionV3

For our project, we observe that the pre-trained model InceptionV3 predicts steroid and Non-steroids images with 97.00% accuracy. For Precision of the Steroid class, we get 0.97 and 0.96 for the Non-steroid class. Next, the Recall of the Steroid class is 0.97, and the Non-steroid class is 0.95. Meanwhile, for the F1-score of the Steroid class is 0.97 and for Non-steroid it is 0.96. The results are found by optimizing our hyperparameters and setting them to the following, Batch of 32, 30 Epochs, Learning rate of 0.0001, Adam optimizer, Activation Function is Sigmoid, and Loss Function is BinaryCross-Entropy Loss.

In Figure 8 below, we can see the Accuracy is 97% for InceptionV3. While, Precision of Steroid class is 0.97 and Non-steroid class is 0.96, Recall of Steroid class is 0.97 and Non-steroid class is 0.95, and F1-score of Steroid class is 0.97 and Non-steroid class is 0.98.

	precision	recall	f1-score	support
non_steroid	0.97	0.97	0.97	279
steroid	0.98	0.98	0.98	392
accuracy			0.98	671
macro avg	0.98	0.97	0.98	671
weighted avg	0.98	0.98	0.98	671

Fig. 8. Classification report of InceptionV3

In Figure 9 below, we observe the training and validation accuracy curve of InceptionV3. we observe the training and validation accuracy curve of InceptionV3 which visualizes the change in Accuracy of Training and Validation on the y-axis over Epoch in the x-axis. From 0 to 5 epoch Validation accuracy and Training accuracy are rising highly and Validation accuracy is higher than training accuracy. From 5 epoch to 30 epoch Validation acccuracy is lower than Training accuracy and both are fairly consistent.

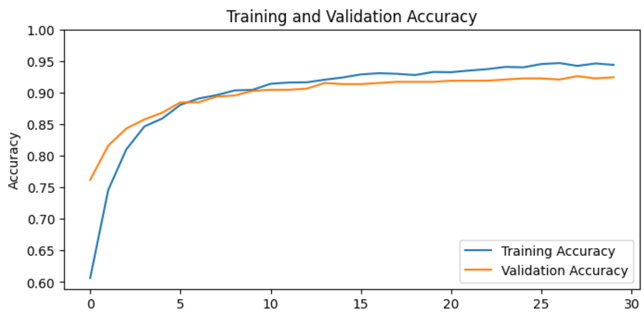

Fig. 9. Training and Validation Accuracy of InceptionV3

In Figure 10 below, we observe the training and validation loss curve of InceptionV3 which visualizes the change in Loss of Training and Validation on the y-axis over Epoch in the x-axis. From 0 to 10 epoch Validation loss and Training loss are dropping highly and Training loss is higher than Validation loss. From 10 epoch to 30 epoch Validation loss and Training loss are both fairly consistent at the same rate.

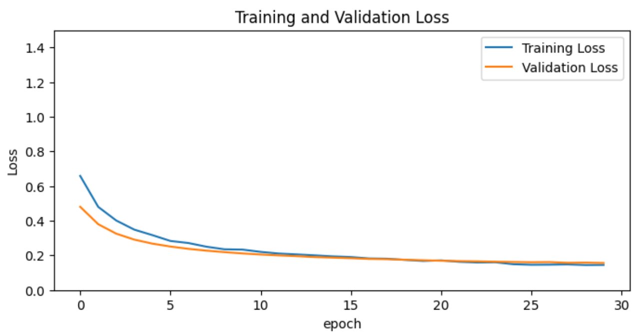

Fig. 10. Training and Validation Loss of InceptionV3

In Figure 11 below, the confusion matrix visualizes all the true and false predictions of steroids and non-steroid classes by InceptionV3. Here, 385 steroid images are predicted as steroid images and 270 nonsteroid images are predicted as nonsteroid images. While, 7 nonsteroid images are predicted wrong and 9 steroid images are predicted wrong too.

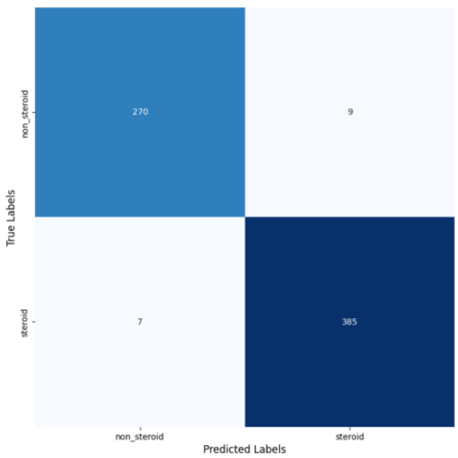

Fig. 11. Confusion Matrix of InceptionV3

C. *EfficientNet-B0*

During our study, we discovered that the pre-trained model EfficientNet-B0 predicts Steroids and Non-steroids images with 97.00% accuracy. In our classification report, we see that the Precision of the Steroid class is 0.97 and the Non-steroid class is 0.96, alternatively, the Recall of the Steroid class is 0.97 and the Non-steroid class is 0.95 lastly, F1-score of the Steroid class is 0.97 and Non-steroid is 0.96. Our study's results are achieved through optimization of our hyperparameter by setting them to the following, Batch of 32, 20 Epoch, Learning rate of 0.0001, Adam optimizer, Activation Function is Sigmoid, and Loss Function is BinaryCross-Entropy Loss.

In Figure 12 below, we can see the Accuracy is 97% for EfficientNet-B0. While, Precision of Steroid class is 0.97 and Non-steroid class is 0.96, Recall of Steroid class is 0.97 and Non-steroid class is 0.95, and F1-score of Steroid class is 0.97 and Non-steroid class is 0.96.

	precision	recall	f1-score	support
non_steroid	0.96	0.95	0.96	279
steroid	0.97	0.97	0.97	392
accuracy			0.97	671
macro avg	0.97	0.96	0.96	671
weighted avg	0.97	0.97	0.97	671

Fig. 12. Classification report of EfficientNet-B0

In Figure 13 below, we observe the training and validation accuracy curve of EfficientNet-B0 which visualizes the change in Accuracy of Training and Validation on the y-axis over Epoch in the x-axis. From 0 to 5 epoch Validation accuracy and Training accuracy are rising highly. From 5 epoch onwards both are fairly consistent. Validation accuracy is higher than training accuracy throughout, suggesting possibility of overfitting.

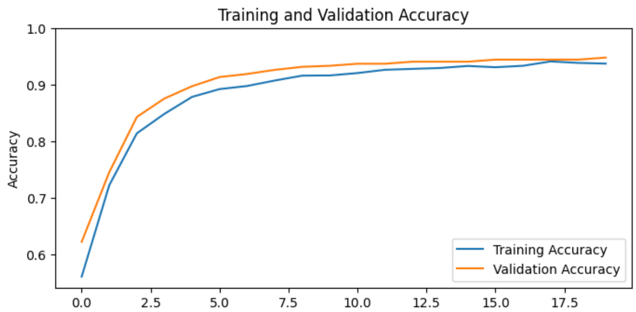

Fig. 13. Training and Validation Accuracy of EfficientNet-B0

In Figure 14 below, we observe the training and validation loss curve of EfficientNet-B0 which visualizes the change in Loss of Training and Validation on the y-axis over Epoch in the x-axis. From 0 to 10 epoch Validation loss and loss are dropping highly. From 10 epoch onwards both are fairly consistent. Validation loss is lower than training loss throughout.

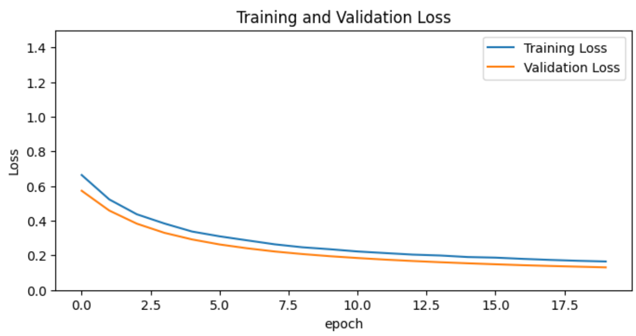

Fig. 14. Training and Validation Loss of EfficientNet-B0

In Figure 15 below, the confusion matrix visualizes all the true and false predictions of steroids and non-steroid classes by EfficientNet-B0. Here, 382 steroid images are predicted as steroid images and 266 nonsteroid images are predicted as nonsteroid images. While, 13 nonsteroid images are predicted wrong and 10 steroid images are predicted wrong too.

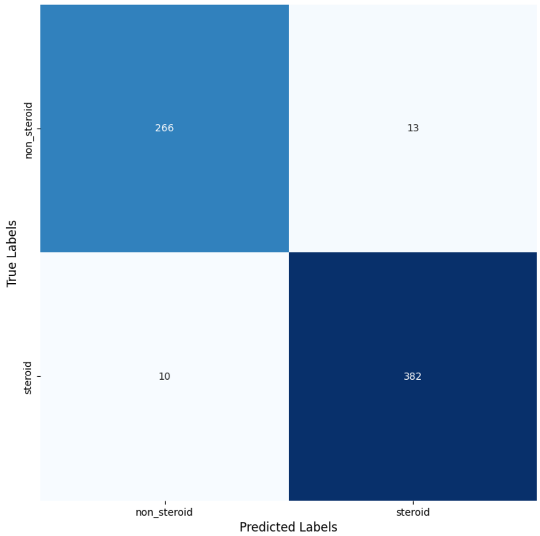

Fig. 15. Confusion Matrix of EfficientNet-B0

D. DenseNet-121

While carrying out our study we saw that pre-trained model DenseNet-121 predicts Steroids and Non-steroids images with 100.00% accuracy. In our classification report, we see that the Precision of the Steroid class is 1.00 and Non-steroid class is also 1.00, in the same way, the Recall of the Steroid class is 1.00 and Non-steroid class is 1.00 lastly, the F1-score of Steroid class is 1.00 and Non-steroid is 1.00. The results were acquired through optimization of our hyperparameter by setting them to the following, Batch of 32, 20 Epochs, Learning rate of 0.0001, Adam optimizer, Activation Function is Softmax, and Loss Function is Cross-Entropy Loss.

In Figure 16 below, we can see the Accuracy is 100% DenseNet-121. While, Precision of Steroid class is 1.00 and Non-steroid class is 1.00, Recall of Steroid class is 1.00 and Non-steroid class is 1.00, and F1-score of Steroid class is 1.00 and Non-steroid class is 1.00.

```
Classification Report:
              precision    recall  f1-score   support

     Steroid       1.00      1.00      1.00       206
  Nonsteroid       1.00      1.00      1.00       215

    accuracy                           1.00       421
   macro avg       1.00      1.00      1.00       421
weighted avg       1.00      1.00      1.00       421
```

Fig. 16. Classification report of DenseNet-121

In Figure 17 below, we observe the training and validation accuracy curve of DenseNet-121 which visualizes the change in Accuracy of Training and Validation on the y-axis over Epoch in the x-axis. From 0 to 12.5 epoch Validation accuracy is consistent with smaller drops, but from 12.5 to 17.5 epoch there is a big drop. For Training accuracy there is a big rise until 2.5 epoch, then a small drop between 2.5 to 7.5 epoch and a bigger drop between 12.5 to 17.5 epoch.

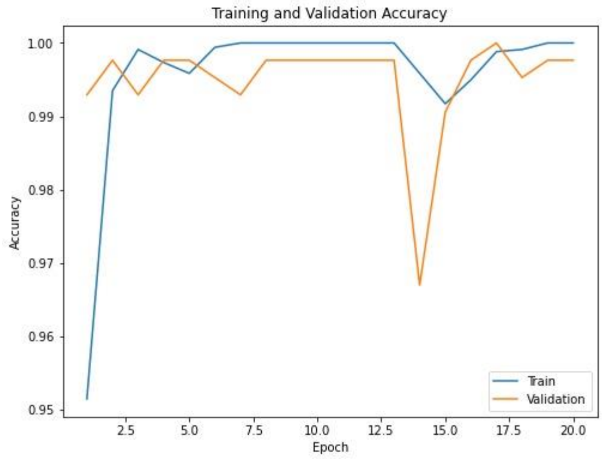

Fig. 17. Block Diagram for Steroid and Non-steroid Image Classification.

In Figure 18 below, we observe the training and validation loss curve of DenseNet-121 which visualizes the change in Loss of Training and Validation on the y-axis over Epoch in the x-axis. From 0 to 2.5 epoch Training loss drops highly and then there is are small rises around 5.0 epoch and 15.0 epoch, otherwise graph stays fairly consistent. For Validation loss it drops until 2.5 epoch and there is a massive rise from 12.5 until 17.5, otherwise graph stays consistent.

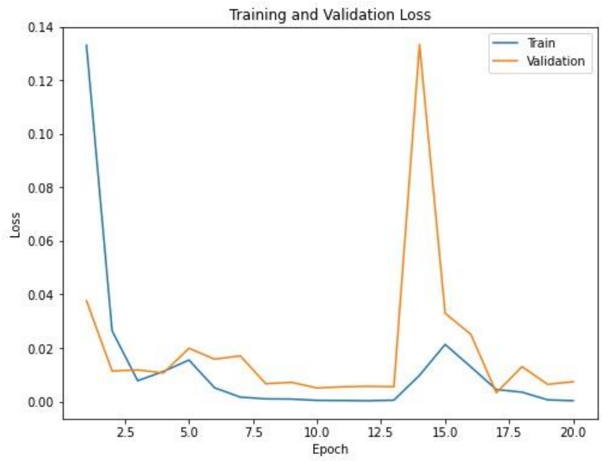

Fig. 18. Training and Validation Loss of DenseNet-121

In Figure 19 below, the confusion matrix visualizes all the true and false predictions of steroids and non-steroid classes by DenseNet-121. Here, 206 steroid images are predicted as steroid images and 215 nonsteroid images are predicted as nonsteroid images. While, 0 nonsteroid and 0 steroid images are predicted wrong.

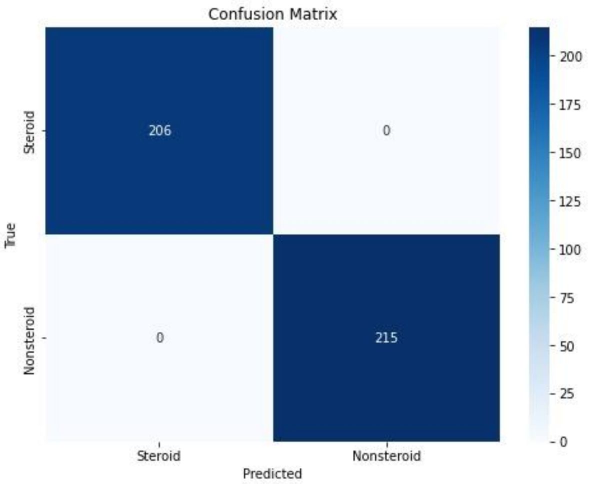

Fig. 19. Training and Validation Loss of DenseNet-121

E.. Hyper-parameters

Hyper-parameter optimization is crucial for optimizing deep learning models. Here we use batch size, epoch, learning rate, optimizer, and loss function as our hyper-parameters [21, 22]. A batch size of 32 is used as a hyper-parameter for our 4 models. Then, we use epoch as one of the main hyper-parameters in the range of 20 to 50 epochs, varying through our 4 models. Next, the learning rate is an important hyper-parameter for training deep learning models because it controls the model's performance. Here, we use 0.01- 0.0001 as our learning rate. After that, we use Adam optimizer because Adam uses adaptive moment estimation of stochastic gradient descent [23]. Here, we use two loss functions, one is cross-entropy loss and another is binary cross-entropy. The difference between Binary cross-entropy loss and cross-entropy loss is in their predicting classes. Binary cross-entropy loss can only deal with the binary classes whereas cross-entropy loss can handle N number of classes. Lastly, we use two activation functions. For binary cross-entropy, we use sigmoid as our activation function on the other hand we use softmax as the activation function for cross-entropy loss.

F. Overall comparison

Table I illustrates the comparison of accuracy, precision, recall and F1-score for 4 models.

TABLE I. MODEL COMPARISON

Model	***Accuracy***	***Precision***		***Recall***	***F1-Score***
Resnet-50	94%	Steroid	0.98	0.89	0.93
		Non-Steroid	0.91	0.98	0.94
EffcientNetB0	97%	Steroid	0.97	0.97	0.97
		Non-Steroid	0.96	0.95	0.96
DenseNet121	100%	Steroid	1.00	1.00	1.00
		Non-Steroid	1.00	1.00	1.00
InceptionV3	98%	Steroid	0.98	0.98	0.98
		Non-Steroid	0.97	0.97	0.97

ACKNOWLEDGMENT

The authors would like to express their heartfelt gratitude towards their project and research supervisor, Dr. Mohammad Monirujjaman Khan, Associate Professor, Department of Electrical and Computer Engineering, North South University, Bangladesh, for his invaluable support, precise guidance, and advice pertaining to the experiments, research and theoretical studies carried out during the course of the current project and also in the preparation of the current report.

Furthermore, the authors would like to thank the Department of Electrical and Computer Engineering, North South University, Bangladesh for facilitating the research. The authors would also like to thank their loved ones for their countless sacrifices and continual support.

REFERENCES

[1] T.I. de Zeeuw, T.M. Brunt, J. van Amsterdam, K. van de Ven, and W. van den Brink, "Anabolic Androgenic Steroid Use Patterns and Steroid Use Disorders in a Sample of Male Gym Visitors," Eur Addict Res, vol. 29, no. 2, pp. 99-108, 2023.

[2] F. Mazzeo, "Anabolic Steroid Use in Sports and in Physical Activity: Overview and Analysis," Sport Mont, vol. 16, pp. 113-118, 2018.

[3] S. S. Liew, M. Khalil-Hani, F. Radzi, and R. Bakhteri, "Gender Classification: A Convolutional Neural Network Approach," Turkish Journal of Electrical Engineering and Computer Sciences, vol. 24, pp. 1248-1264, 2016.

[4] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," Global Transitions Proceedings, vol. 3, no. 1, pp. 305-310, 2022.

[5] M. J. Yousif, “Enhancing the accuracy of image classification using Deep Learning and preprocessing methods,” *Artificial Intelligence & Robotics Development Journal*, Jan. 2024.

[6] S.-T. Bow, *Pattern Recognition and Image Preprocessing*. New York, United States of America: Marcel Dekker, 2002.

[7] X. Han et al., “Pre-trained models: Past, present and future,” arXiv.org, https://arxiv.org/abs/2106.07139 (accessed Mar. 27, 2024).

[8] X. Qiu et al., “Pre-trained models for Natural Language Processing: A Survey - Science China Technological Sciences,” SpringerLink, https://link.springer.com/article/10.1007/s11431-020-1647-3 (accessed Mar. 27, 2024).

[9] J. Gu et al., “Recent advances in Convolutional Neural Networks,” arXiv.org, https://arxiv.org/abs/1512.07108 (accessed Mar. 27, 2024).

[10] L. Alzubaidi *et al.*, “Review of Deep Learning: Concepts, CNN Architectures, challenges, applications, Future Directions,” *Journal of Big Data*, vol. 8, no. 1, Mar. 2021.

[11] A. Sarkar, “Creating densenet 121 with TensorFlow,” Medium, https://towardsdatascience.com/creating-densenet-121-with-tensorflow-edbc08a956d8 (accessed Mar. 27, 2024).

[12] A. Qamar Bhatti *et al.*, “Explicit content detection system: An approach towards a safe and ethical environment,” *Applied Computational Intelligence and Soft Computing*, vol. 2018, pp. 1–13, Jul. 2018.

[13] S. A. Agrawa et al., “International Journal of Intelligent Systems and Applications,” International Journal of Intelligent Systems and Applications(IJISA), https://www.mecs-press.org/ijisa/ (accessed Mar. 27, 2024).

[14] C. Peng, Y. Liu, X. Yuan, and Q. Chen, "Research of image recognition method based on enhanced Inception-ResNet-V2," Multimedia Tools and Applications, vol. 81, pp. 1-21, 2022.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818-2826, 2016

[16] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019.

[17] P. Potrimba, “What is EfficientNet? The Ultimate Guide.,” Roboflow Blog, https://blog.roboflow.com/what-is-efficientnet/ (accessed Mar. 27, 2024).

[18] N. Ali, “Creating datasets for machine learning. my own lessons and experience.,” Medium, https://medium.com/swlh/creating-datasets-for-machine-learning-my-own-lessons-and-experience-44e118b15d6a (accessed Mar. 27, 2024).

[19] A. Venzke, D. K. Molzahn, and S. Chatzivasileiadis, “Efficient creation of datasets for data-driven power system applications,” *Electric Power Systems Research*, vol. 190, p. 106614, Jan. 2021.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[21] R. Kumar, M. Sharma, K. Dhawale, and G. Singal, "Identification of Dog Breeds Using Deep Learning," pp. 193-198, 2019.